KERNEL MULTI-METRIC LEARNING FOR MULTI-CHANNEL TRANSIENT ACOUSTIC SIGNAL CLASSIFICATION

Haichao Zhang[†‡], *Yanning Zhang*[†*], *Nasser M. Nasrabadi*[§], *and Thomas S. Huang*[‡]

[†] School of Computer Science, Northwestern Polytechnical University, Xi'an, China 710129
[‡] Beckman Institute, University of Illinois at Urbana-Champaign, IL 61801
[§] US Army Research Laboratory, 2800 Powder Mill Road, Adelphi, MD 20783

ABSTRACT

In this paper, we propose a kernel multi-metric learning algorithm for multi-channel transient acoustic signal classification. The proposed method learns a set of metrics jointly for multi-channel transient acoustic signals in a kernel-induced feature space to exploit the non-linearity of the data for improving the classification performance. An effective algorithm is developed for the task of learning multiple metrics in the kernel space. By learning the multiple metrics jointly within a single unified optimization framework, we can learn better metrics to integrate the multiple channels of the signal for a joint classification. Experimental results compared with classical as well as recent algorithms on real-world acoustic datasets verified the effectiveness of the proposed method.

***Index Terms*—** metric learning, kernel learning, multi-channel acoustic signal classification

1. INTRODUCTION

Transient acoustic signal classification is an important topic in surveillance and security. It applications range from daily life to battlefield tasks [1]. The challenge of transient acoustic signal classification lies in the fact that the typical environment is not ideal, but is usually noisy with environmental variations. To handle the noise and extract useful features for classification, variant techniques have been proposed [1, 2, 3]. In [1], a maximum likelihood method was proposed for restoring transient signals from a sensor network with wavelet subband features for classification. The authors of [2] proposed a denoising technique based on short time spectral attenuation for signals from a microphone array for target detection and localization. In [3], a wavelet packet transformation was adopted for feature extraction followed by classification. Almost all the previous algorithms on acoustic signal classification have ignored the use of multiple measurements as in the case of multi-channel signals for improving the classification performance.

*Corresponding author

In our previous work [4], a heterogeneous multi-metric learning (HMML) method for multi-channel transient acoustic signal classification is developed. While applied onto only multi-channel acoustic signals in [4], the algorithm can be potentially applied to signals collected from heterogeneous sources. In this work, we extend our previous work and propose a multi-metric learning algorithm in the Reproducing Kernel Hilbert Space (RKHS), which can exploit the non-linearity of the data in the feature space via a non-linear mapping associated with a kernel. Experimental results verified the effectiveness of the proposed method over several conventional and comparable methods.

2. HETEROGENEOUS MULTI-METRIC LEARNING MODEL REVISITED

The aim of HMML is to learn a projection set $\{\mathbf{P}^s\}_{s=1}^S$ (or a metric set $\{\mathbf{M}^s\}_{s=1}^S$, where $\mathbf{M}^s = \mathbf{P}^{s\top}\mathbf{P}^s$) adapted to each channel for improving the joint classification performance. Given N training samples from S potentially heterogeneous channels $\left\{(\{\mathbf{x}_i^s\}_{s=1}^S, y_i)\right\}_{i=1}^N$, the following model is used in [4] to learn the metric set,

$$\text{Minimize} \quad E\big(\{\mathbf{P}^s\}_{s=1}^S\big) = \quad (1-\lambda)E_{\text{pull}} + \lambda E_{\text{push}}, \tag{1}$$

where $\mathbf{P}^s$ is the projection matrix for the s-th channel, and

$$E_{\text{pull}}(\{\mathbf{P}^s\}_{s=1}^S) = \sum_{i,j\rightsquigarrow i}\sum_{s=1}^S \left\|\mathbf{P}^s(\mathbf{x}_i^s - \mathbf{x}_j^s)\right\|^2,$$

$$E_{\text{push}}(\{\mathbf{P}^s\}_{s=1}^S) = \sum_{i,j\rightsquigarrow i}\sum_l (1-y_{il})\Big[1 + \sum_{s=1}^S \left\|\mathbf{P}^s(\mathbf{x}_i^s - \mathbf{x}_j^s)\right\|^2 - \sum_{s=1}^S \|\mathbf{P}^s(\mathbf{x}_i^s - \mathbf{x}_l^s)\|^2\Big]_+,$$

where i and l are indexes of the training samples and $j \rightsquigarrow i$ denotes the set of "target" neighbors of $\mathbf{x}_i$, *i.e.*, the k nearest samples with the same label as $\mathbf{x}_i$. $y_{il} \in \{0,1\}$ is a binary number indicating whether $\mathbf{x}_i$ and $\mathbf{x}_l$ are of the same class. $[\cdot]_+ = \max(\cdot, 0)$ is a hinge loss. The samples contributing to the energy $E_{\text{push}}(\mathbf{P})$ are termed as "impostors," which are those samples within the radius defined by target samples

(plus a margin) but belong to classes which is different from the target class.

To solve (1) effectively, a gradient-based optimization algorithm is developed in [4]. After the metric set $\{\mathbf{M}^s\}_{s=1}^S$ is learned, we can proceed to perform classification by integrating the information from all the channels. Given a multi-channel test sample $\mathbf{x}_t = \{\mathbf{x}_t^s\}_{s=1}^S$, we can classify it using the following energy-based classification method given by (3), which can be used for better classification performance [5]. Denoting the distance between the multi-channel test sample $\mathbf{x}_t$ and a multi-channel training sample $\mathbf{x}_i = \{\mathbf{x}_i^s\}_{s=1}^S$ as

$$D_{\mathbf{M}}(\mathbf{x}_t, \mathbf{x}_i) = \sum_{s=1}^{S} d_{\mathbf{M}^s}(\mathbf{x}_t^s, \mathbf{x}_i^s), \tag{2}$$

then the energy-based classification can be achieved via [5]:

$$\begin{aligned}\hat{y}_t = \arg\min_{y_t} & (1-\lambda)\sum_{j \rightsquigarrow t} D_{\mathbf{M}}(\mathbf{x}_t, \mathbf{x}_j) \\ & + \lambda \sum_{j \rightsquigarrow t,l} (1-y_{tl}) \Big[1 + D_{\mathbf{M}}(\mathbf{x}_t, \mathbf{x}_j) - D_{\mathbf{M}}(\mathbf{x}_t, \mathbf{x}_l)\Big]_+ \\ & + \lambda \sum_{i,j \rightsquigarrow i} (1-y_{it}) \Big[1 + D_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) - D_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_t)\Big]_+ .\end{aligned} \tag{3}$$

The first term in (3) represents the accumulated energy for the k target neighbors of $\mathbf{x}_t$; the second term accumulates the hinge loss over all the imposters for $\mathbf{x}_t$; the third term represents the accumulated energy for different labeled samples whose neighbor perimeters are invaded by $\mathbf{x}_t$, *i.e.*, taking $\mathbf{x}_t$ as their imposter.

3. MULTI-METRIC LEARNING IN THE REPRODUCING KERNEL HILBERT SPACE

In this section, we present a multi-metric learning method in high-dimensional feature space induced by kernel mapping as a generalization of HMML method in [4]. We denote this method as kernel-based HMML (KHMML) in the sequel.

3.1. Multi-Metric Learning in Kernel Space

By introducing a non-linear feature mapping function $\phi(\cdot) : \mathbb{R}^m \to \mathbb{R}^n$ with $n \gg m$, and denote $\phi_i^s = \phi(\mathbf{x}_i^s)$, we can formulate the KHMML model as follows:

$$\text{Minimize} \quad \mathcal{E}\big(\{\mathbf{P}^s\}_{s=1}^S\big) = (1-\lambda)\mathcal{E}_{\text{pull}} + \lambda\mathcal{E}_{\text{push}}, \tag{4}$$

where $\mathbf{P}^s$ is the projection matrix for the s-th channel in the kernel space, and

$$\mathcal{E}_{\text{pull}}(\{\mathbf{P}^s\}_{s=1}^S) = \sum_{i,j \rightsquigarrow i} \sum_{s=1}^{S} \big\|\mathbf{P}^s(\phi_i^s - \phi_j^s)\big\|^2,$$

$$\begin{aligned}\mathcal{E}_{\text{push}}(\{\mathbf{P}^s\}_{s=1}^S) = \sum_{i,j \rightsquigarrow i} \sum_{l} (1-y_{il}) \Big[1 + \sum_{s=1}^{S} \big\|\mathbf{P}^s(\phi_i^s - \phi_j^s)\big\|^2 \\ - \sum_{s=1}^{S} \|\mathbf{P}^s(\phi_i^s - \phi_l^s)\|^2\Big]_+ .\end{aligned}$$

By differentiating (4) with respect to $\mathbf{P}^s$, we get the following expression:

$$\mathbf{Q}_t^s = \frac{\partial \mathcal{E}(\{\mathbf{P}_t^s\}_{s=1}^S)}{\partial \mathbf{P}_t^s} \tag{5}$$

$$\begin{aligned}&= (1-\lambda)\mathbf{P}_t^s \sum_{i,j \rightsquigarrow i} (\phi_i^s - \phi_j^s)(\phi_i^s - \phi_j^s)^\top \\ &+ \lambda \mathbf{P}_t^s \sum_{(i,j,l) \in \mathcal{N}_t} [(\phi_i^s - \phi_j^s)(\phi_i^s - \phi_j^s)^\top - (\phi_i^s - \phi_l^s)(\phi_i^s - \phi_l^s)^\top],\end{aligned}$$

where $\mathcal{N}_t$ is defined as the set of triple-indices $(i,j,l) \in \mathcal{N}_t$ if and only if (i,j,l) triggers the hinge loss in $\mathcal{E}_{\text{push}}$. Note that in this case, as the dimensionality of RKHS induced by $\phi(\cdot)$ may be infinite, it is not possible to update the projection set $\{\mathbf{P}^s\}$ directly. Therefore, to learn the projection set $\{\mathbf{P}^s\}$ in the kernel space, we adopt a parametric representation for it as a linear combination of the feature vectors in the form of $\mathbf{P}^s = \Theta^s \mathbf{\Phi}^{s\top}$, where Θ^s is referred to as the combination coefficient matrix and $\mathbf{\Phi}^s$ denotes the data matrix in RKHS constructed from N training samples for channel s as $\mathbf{\Phi}^s = \big[\phi_1^s, \phi_2^s, \cdots, \phi_N^s\big]$. Then the projection of a sample $\mathbf{x}_t^s$ with $\mathbf{P}^s$ in the RKHS can be computed as:

$$\mathbf{P}^s \phi(\mathbf{x}_t^s) = \Theta \mathbf{\Phi}^{s\top} \phi_t^s = \Theta \begin{bmatrix} k(\mathbf{x}_1^s, \mathbf{x}_t^s) \\ k(\mathbf{x}_2^s, \mathbf{x}_t^s) \\ \vdots \\ k(\mathbf{x}_N^s, \mathbf{x}_t^s) \end{bmatrix} = \Theta \mathbf{k}_t^s, \tag{6}$$

where $k(\cdot,\cdot)$ is the kernel function associated with the feature mapping function $\phi(\cdot)$, specifically, $k(\mathbf{x}_1, \mathbf{x}_2) = \phi(\mathbf{x}_1)^\top \phi(\mathbf{x}_2)$. Substituting $\mathbf{P}^s = \Theta^s \mathbf{\Phi}^{s\top}$ into (5) and using (6), we get:

$$\mathbf{Q}_t^s = (1-\lambda)\Theta_t^s \mathbf{\Phi}^{s\top} \textstyle\sum_{i,j \rightsquigarrow i} (\phi_i^s - \phi_j^s)(\phi_i^s - \phi_j^s)^\top \tag{7}$$

$$\begin{aligned}&+ \lambda \Theta_t^s \mathbf{\Phi}^{s\top} \sum_{(i,j,l) \in \mathcal{N}_t} \big[(\phi_i^s - \phi_j^s)(\phi_i^s - \phi_j^s)^\top - (\phi_i^s - \phi_l^s)(\phi_i^s - \phi_l^s)^\top\big] \\ &= (1-\lambda)\Theta_t^s \sum_{i,j \rightsquigarrow i} (\mathbf{k}_i^s - \mathbf{k}_j^s)(\phi_i^s - \phi_j^s)^\top \\ &+ \lambda \Theta_t^s \sum_{(i,j,l) \in \mathcal{N}_t} \big[(\mathbf{k}_i^s - \mathbf{k}_j^s)(\phi_i^s - \phi_j^s)^\top - (\mathbf{k}_i^s - \mathbf{k}_l^s)(\phi_i^s - \phi_l^s)^\top\big].\end{aligned}$$

Note that for the term $(\mathbf{k}_i^s - \mathbf{k}_j^s)(\phi_i^s - \phi_j^s)^\top$, we can reformulate it as follows:

$$\begin{aligned}&(\mathbf{k}_i^s - \mathbf{k}_j^s)(\phi_i^s - \phi_j^s)^\top = (\mathbf{k}_i^s - \mathbf{k}_j^s)\phi_i^{s\top} - (\mathbf{k}_i^s - \mathbf{k}_j^s)\phi_j^{s\top} \\ &= \mathbf{\Delta}_i^{(\mathbf{k}_i^s - \mathbf{k}_j^s)} \mathbf{\Phi}^{s\top} - \mathbf{\Delta}_j^{(\mathbf{k}_i^s - \mathbf{k}_j^s)} \mathbf{\Phi}^{s\top} \\ &= [\mathbf{\Delta}_i^{(\mathbf{k}_i^s - \mathbf{k}_j^s)} - \mathbf{\Delta}_j^{(\mathbf{k}_i^s - \mathbf{k}_j^s)}] \mathbf{\Phi}^{s\top},\end{aligned} \tag{8}$$

where $\mathbf{\Delta}_i^{(\mathbf{x})}$ is a matrix constructed by using $\mathbf{x}$ as its i-th column vector and zeros elsewhere:

$$\mathbf{\Delta}_i^{(\mathbf{x})} = \Big[\overbrace{\mathbf{0}, \cdots, \mathbf{0}}^{i-1 \text{ columns}}, \mathbf{x}, \mathbf{0}, \cdots, \mathbf{0}\Big]. \tag{9}$$

Substituting (8) into (7), we can get:

$$\begin{aligned}\mathbf{Q}_t^s &= (1-\lambda)\Theta_t^s \sum_{i,j\rightsquigarrow i}\left[\mathbf{\Delta}_i^{(\mathbf{k}_i^s-\mathbf{k}_j^s)} - \mathbf{\Delta}_j^{(\mathbf{k}_i^s-\mathbf{k}_j^s)}\right]\mathbf{\Phi}^{s\top}\\ &+ \lambda\Theta_t^s \sum_{(i,j,l)\in\mathcal{N}_t}\left[\mathbf{\Delta}_i^{(\mathbf{k}_i^s-\mathbf{k}_j^s)} - \mathbf{\Delta}_j^{(\mathbf{k}_i^s-\mathbf{k}_j^s)} - \mathbf{\Delta}_i^{(\mathbf{k}_i^s-\mathbf{k}_l^s)} + \mathbf{\Delta}_l^{(\mathbf{k}_i^s-\mathbf{k}_l^s)}\right]\mathbf{\Phi}^{s\top}\\ &= \Omega^s\mathbf{\Phi}^{s\top},\end{aligned}$$

where $\Omega^s = (1-\lambda)\Theta_t^s\sum_{i,j\rightsquigarrow i}\left[\mathbf{\Delta}_i^{\mathbf{k}_i^s-\mathbf{k}_j^s} - \mathbf{\Delta}_j^{\mathbf{k}_i^s-\mathbf{k}_j^s}\right] + \lambda\Theta_t^s\sum_{(i,j,l)\in\mathcal{N}_t}\left[\mathbf{\Delta}_i^{\mathbf{k}_i^s-\mathbf{k}_j^s} - \mathbf{\Delta}_j^{\mathbf{k}_i^s-\mathbf{k}_j^s} - \mathbf{\Delta}_i^{\mathbf{k}_i^s-\mathbf{k}_l^s} + \mathbf{\Delta}_l^{\mathbf{k}_i^s-\mathbf{k}_l^s}\right]$. By the above derivation, we have represented the kernel gradient direction as a linear function of the kernel matrix. Therefore, we can represent the updated projection $\mathbf{P}^s$ by using the updated combination coefficient matrix Θ^s at time step $t+1$ as:

$$\begin{aligned}\mathbf{P}_{t+1}^s \leftarrow \mathbf{P}_t^s - \alpha\mathbf{Q}_t^s &= \Theta_t^s\mathbf{\Phi}^{s\top} - \alpha\Omega^s\mathbf{\Phi}^{s\top}\\ &= (\Theta_t^s - \alpha\Omega^s)\mathbf{\Phi}^{s\top} = \Theta_{t+1}^s\mathbf{\Phi}^{s\top}.\end{aligned} \tag{10}$$

Therefore, by (10), we have learned the projection matrix $\mathbf{P}^s$ in RKHS, which is directly intractable. The above derivation is inspired by [6], which is designed for single metric learning. The major steps of the proposed KHMML algorithm is summarized in Algorithm 1.

3.2. Multi-Channel Signal Classification in Kernel Space

After we learn the combination coefficient matrix set $\{\Theta^s\}_{s=1}^S$, we can use it to classify the test samples. To do that, we first show how to calculate the distance in the kernel space with the learned kernel metric set. The distance between the i-th and j-th samples can be calculated as:

$$\begin{aligned}d_{\mathbf{M}^s}(\phi(\mathbf{x}_i^s),\phi(\mathbf{x}_j^s)) &= \|\mathbf{P}^s\phi(\mathbf{x}_i^s) - \mathbf{P}^s\phi(\mathbf{x}_j^s)\|_2^2\\ &= \|\Theta^s(\mathbf{k}_i^s - \mathbf{k}_j^s)\|_2^2.\end{aligned} \tag{11}$$

By substituting this into (2) and (3), we can perform classification for the test sample with the learned metric in RKHS.

Algorithm 1: Kernel Multi-Metric Learning.

Input: training set $\{(\{\mathbf{x}_i^s\}_{s=1}^S, y_i)\}_{i=1}^N$, number of nearest neighbor L, kernel function $k(\cdot,\cdot)$
Output: combination coefficient matrix set $\{\Theta^s\}_{s=1}^S$ used for multi-metrics in kernel space
Initialize: $t \leftarrow 0$, $\{\Theta_t^s\}_{s=1}^S, \mathcal{N}_t = \{\}$;
while *convergence condition false* **do**
  Update the active set $\mathcal{N}_{t+1}$ by collecting the triplets (i,j,l) that incur the hinge loss in kernel space;
  for $s = 1, 2, \cdots, S$ **do**
    Compute the gradient Ω^s for the s-th channel;
    Take gradient step for the combination coefficient matrix of the s-th channel:
    $\Theta_{t+1}^s \leftarrow \Theta_t^s - \alpha\Omega^s$;
  $t \leftarrow t+1$

Table 1. Classification accuracy for the two-class mortar problem ($S = 4, L = 3, r = 0.5$).

Dataset	1	2	3	4	Average
Logistic	0.7778	0.8069	0.7183	0.6857	0.7472
SVM	0.8073	0.7991	0.7917	0.7693	0.7919
CSVM	0.8173	0.8448	0.7938	0.8000	0.8140
HMML [4]	0.8673	0.8621	0.8525	0.8240	0.8515
JSRC [7]	0.8515	**0.8828**	**0.8857**	0.8147	0.8534
KHMML	**0.8728**	**0.8828**	0.8607	**0.8360**	**0.8631**

4. EXPERIMENT RESULTS

In this section, we carry out experiments on a number of real acoustic datasets and compare the results with several conventional classification methods to verify the effectiveness of the proposed method. We use the multi-channel transient acoustic dataset collected for launch and impact of different weapons (mortar and rocket) using a tetrahedral acoustic sensor array. For each event, the acoustic sensor array measures the signal from a launch/impact event using four acoustic sensors simultaneously. We have a total of four datasets (referred to as Dataset 1∼4) [7]. Among these four datasets, some consist of four subsets collected by an acoustic sensor array deployed at four different physical sites. We first segment the raw signal with spectral maximum detection [8] in order to locate the physical event and then extract the first 50 Cepstral coefficients (start from the second coefficient) [9] for classification.

To evaluate the effectiveness of the proposed method, we compare the results with different classical algorithms, including sparse linear multinomial Logistic Regression [10] and Linear Support Vector Machine (SVM) [11], which is used in two modes in our experiments: (1) treating each sensor signal separately (SVM); (2) concatenating all the signals from different sensors (CSVM). One-vs.-all scheme is used for SVM in the case of multi-class classification. The joint sparse representation-based classification method (JSRC) [7] and our previously proposed HMML method [4] are also compared. For KHMML, Gaussian kernel is used in our experiments, with bandwidth $\sigma = 0.8$, which gives desirable results empirically. The number of nearest neighbors is set as $L = 3$. The combination weight is set as $\lambda = 0.5$.

4.1. Two-Class Event Classification

In this experiment, we focus on the classification problem between launch and impact for a single kind of weapon (mortar) using all four datasets. We randomly split each dataset into two halves (training ratio $r = 0.5$) for training and testing and run the experiment five times. We report the average performance in Table 1 for the four datasets. It can be seen that KHMML outperforms HMML for all the datasets, indicating

Table 2. Classification accuracy for the four-class problem ($S = 4, L = 3, r = 0.5$).

Dataset	1	2	3	4	Average
Logistic	0.7440	0.7234	0.6882	0.7367	0.7231
SVM	0.7410	0.7227	0.6860	0.7474	0.7243
CSVM	0.7487	0.7375	0.6945	0.7169	0.7244
KNN	0.6204	0.7188	0.6236	0.7456	0.6771
HMML [4]	0.8014	0.7313	0.7284	0.7928	0.7635
JSRC [7]	0.8152	**0.7969**	0.7494	0.7928	0.7886
KHMML	**0.8252**	0.7862	**0.7512**	**0.8632**	**0.8065**

Table 3. Classification accuracy for the four-class classification with training and testing on data measured at different physical sites ($S = 4, L = 3$).

Method	Physical Sites				Average
	Site 1	Site 2	Site 3	Site 4	
Logistic	0.6797	0.6829	0.7314	0.7109	0.6394
SVM	0.6901	0.7134	0.8005	0.6652	0.6449
CSVM	0.7292	0.7073	0.7766	0.7043	0.6574
HMML [4]	0.7917	0.7805	0.7766	0.7391	0.7627
JSRC [7]	0.8125	0.8049	0.7447	**0.7652**	0.7818
KHMML	**0.8229**	**0.8537**	**0.8298**	0.7565	**0.8157**

the effectiveness of the multi-metric learning in kernel space for exploiting the non-linearity over the linear property. Furthermore, the proposed KHMML method performs overall better than the joint sparse representation-based method [7], which has been shown to be effective for the multi-channel transient acoustic signal classification task.

4.2. Four-Class Event Classification

To further verify the effectiveness of the proposed method, we test it on a four-class classification problem, where we want to decide whether the event is launch or impact and whether the weapon is mortar or rocket, which is much more challenging. Similarly, we generate training and testing datasets by randomly splitting each dataset into two halves. We repeat the experiment five times and report the average performance for each dataset as well as the overall average classification accuracy in Table 2. Again, it is shown that the KHMML method improves the classification accuracy over HMML by a large margin on all the datasets, and performs comparable to or better than JSRC on different datasets.

4.3. Considering the Effects of Sensor Sites

In this experiment, to investigate the classification performance using data captured by sensors at different physical sites, we generate training and testing datasets according to the physical sites where the acoustic sensor array is deployed. Specifically, the Dataset 2 contains subsets collected from four different sites. We keep all the data from one site for testing and data from all the other sites for training; we do this for each dataset. The classification results are summarized in Table 3. Table 3 shows that the proposed KHMML method is more robust to sensors' site locations and performs the best on average, indicating the effectiveness brought by joint multi-metric learning in the kernel space.

5. CONCLUSIONS

We have presented in this paper an effective method to learn jointly a set of metrics in the kernel space for multi-channel transient acoustic signal classification. By exploiting the non-linearity in the data via kernel mapping, we are able to learn a set of metrics adapted for each channel jointly for improving the classification performance. Experiments on real-world multi-sensor datasets compared with several conventional as well as recent developed methods verified the effectiveness of the proposed method. The method developed in this paper is not limited to acoustic signals and is readily applicable to other classification tasks.

Acknowledgement We would like to thank all the reviewers for their reconstructive comments. This work is supported by NSF (60872145, 60903126), National High-Tech.(2009AA01Z315), Postdoctoral Science Foundation (20090451397, 201003685), Cultivation Fund from Ministry of Education (708085) of China and U.S. Army Research Laboratory and U.S. Army Research Office under grant number W911NF-09-1-0383.

6. REFERENCES

[1] M.R. Azimi-Sadjadi, Y. Yang, and S. Srinivasan, "Acoustic classification of battlefield transient events using wavelet sub-band features," in *Proc. SPIE'07 Defense and Security Symposium*, 2007, vol. 6562.

[2] D. Salvati and S. Canazza, "Improvement of acoustic localization using a short time spectral attenuation with a novel suppression rule," in *Proc. of the 12th Int. Conference on Digital Audio Effects (DAFx-09)*, 2009.

[3] R.E. Learned and A.S. Willsky, "A wavelet-packet approach to transient signal classification," *Applied Computational and Harmonic Analysis*, pp. 265–278, 1995.

[4] H. Zhang, N. Nasrabadi, T. Huang, and Y. Zhang, "Heterogeneous multi-metric learning for multi-sensor fusion," in *International Conference on Information Fusion*, 2011, pp. 1–8.

[5] K. Q. Weinberger, J. Blitzer, and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," in *NIPS*, 2006, pp. 1473–1480.

[6] L. Torresani and K. Lee, "Large margin component analysis," in *NIPS*, 2007, pp. 1385–1392.

[7] H. Zhang, N. Nasrabadi, T. Huang, and Y. Zhang, "Transient acoustic signal classification using joint sparse representation," in *ICASSP*, 2011, pp. 2220–2223.

[8] S. Engelberg and E. Tadmor, "Recovery of edges from spectral data with noise-a new perspective," *SIAM Journal on Numerical Analysis*, vol. 46, no. 5, pp. 2620–2635, 2008.

[9] D. G. Childers, D. P. Skinner, and R. C. Kemerait, "The cepstrum: A guide to processing," in *Proc. IEEE*, 1977, vol. 65, pp. 1428–1443.

[10] P. Tseng and S. Yun, "A coordinated gradient descent method for nonsmooth separable minimizatin," *Math. Prog.*, vol. 117, pp. 387–423, 2009.

[11] C. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.